

NOEH, Noah-Vincenz (nn4718)



531 tk106 5
a5 nn4718 v1



Electronic submission



Thu - 22 Nov 2018 17:48:00

nn4718

Exercise Information

Module: 531 Prolog
Exercise: 5 (LAB)
Title: Syllogisms
FAO: Kimber, Timothy (tk106)

Issued: Wed - 14 Nov 2018
Due: Thu - 22 Nov 2018
Assessment: Individual
Submission: Electronic

Student Declaration - Version 1

- I declare that this final submitted version is my unaided work.

Signed: (electronic signature) Date: 2018-11-22 17:47:28

For Markers only: (circle appropriate grade)

NOEH, (nn4718)	Noah-Vincenz	01562775	a5	2018-11-22 17:47:28	A*	A	B	C	D	E	F
-------------------	--------------	----------	----	---------------------	----	---	---	---	---	---	---

Prolog Syllogisms/TestSummary

TestSummary.txt: 1/1

Noah-Vincenz Noeh - nn4718: a5

```
1: Prolog Syllogisms: Summary for nn4718 of a5
2: -----
3:
4:   Full Tests:
5:     Step 2 (syllogism/1):          38 / 38
6:     Step 3 (translate/1, cl/3):    12 / 12
7:     Step 4 (eval/2, valid/1, invalid/1): 13 / 17
8:     Step 5 (test/1):              8 / 8
9:
```

```

1: %% File: syllogisms.pl
2: %% Name: Noah-Vincenz Noeh
3: %% Date: 15/11/2018
4: %%
5: %% This program is a solution to Prolog 531 Assessed Exercise 5 'Syllogisms'
6: %% The exercise is to develop a parser and meta-interpreter for syllogistic
7: %% sentences, and use these to build a tool to determine the validity of a
8: %% syllogistic argument.
9:
10: %% ----- Step 1 -----%%
11:
12: %% opposite(+L, -Opp)
13: :- include(arguments).
14: :- include(utilities).
15: opposite([no|Tail], [some|Tail]) :- !.
16: opposite([some, Thing1, is, not, a, Thing2], [a, Thing1, is, a, Thing2]) :- !.
17: opposite([some, Thing1, is, not, Thing2], [a, Thing1, is, Thing2]) :- !.
18: opposite([some|Tail], [no|Tail]) :- !.
19: % for 'a' and 'every'
20: opposite([_, Thing1, is, a, Thing2], [some, Thing1, is, not, a, Thing2]).
21: opposite([_, Thing1, is, Thing2], [some, Thing1, is, not, Thing2]).
22:
23:
24: %% ----- Step 2 -----%%
25:
26: %% Stage 2.1 - This is the suggested way to develop the solution.
27: %% Once Stage 2.2 is complete you can delete or comment out this code.
28: %% syllogism/0
29:
30: /*
31: syllogism --> article, [B], is_, optional_article, [C].
32: syllogism --> some, [B], is_, not, optional_article, [C].
33: syllogism --> no_, [B], is_, optional_article, [C].
34: syllogism --> some, [B], is_, optional_article, [C].
35: article --> [a].
36: article --> [every].
37: some --> [some].
38: no_ --> [no].
39: is_ --> [is].
40: not --> [not].
41: optional_article --> [a].
42: optional_article --> [].
43: */
44:
45: %% Stage 2.2
46: %% syllogism(-Clauses)
47:
48: syllogism(Clauses) --> article, [B], is_, optional_article, [C], {
49:     Term1 =.. [C, X],
50:     Term2 =.. [B, X],
51:     Clauses = [(Term1:-Term2)]
52: }.
53:
54: syllogism(Clauses) --> some, [B], is_, not, optional_article, [C], {
55:     Term1 =.. [not, C],
56:     Term2 =.. [some, B, Term1],
57:     Term3 =.. [B, Term2],
58:     Term4 = (Term3:-true),
59:     Term5 =.. [C, Term2],
60:     Term6 = (false:-Term5),
61:     Clauses = [Term4,Term6]
62: }.
63:
64: syllogism(Clauses) --> no_, [B], is_, optional_article, [C], {
65:     Term1 =.. [B, X],
66:     Term2 =.. [C, X],
67:     Clauses = [(false:-Term1, Term2)]

```

```

68: }.
69:
70: syllogism(Clauses) --> some, [B], is_, optional_article, [C], {
71:     Term1 =.. [some, B, C],
72:     Term2 =.. [B, Term1],
73:     Term3 = (Term2:-true),
74:     Term4 =.. [C, Term1],
75:     Term5 = (Term4:-true),
76:     Clauses = [Term3, Term5]
77: }.
78:
79: article --> [a].
80: article --> [every].
81: some --> [some].
82: no_ --> [no].
83: is_ --> [is].
84: not --> [not].
85: optional_article --> [a].
86: optional_article --> [].
87:
88:
89: %% ----- Step 3 -----%%
90:
91: %% translate(+N)
92: %:- dynamic cl/3.
93: translate(N) :-
94:     findall(Premise, p(N, Premise), List),
95:     write(List),
96:     nl,
97:     c(N, Conclusion),
98:     opposite(Conclusion, Opposite),
99:     get_clauses([Opposite|List], [], Phrases),
100:     write(Phrases),
101:     assertall(N, Phrases).
102:
103: get_clauses([], Acc, Acc).
104: get_clauses([X|Tail], Acc, Phrases) :-
105:     phrase(syllogism(Clauses), X), % returns semantics as clauses
106:     append(Clauses, Acc, NewAcc),
107:     get_clauses(Tail, NewAcc, Phrases).
108:
109: %% ----- Step 4 -----%%
110:
111: %% eval(+N, +Calls)
112: eval(_, true) :- !.
113: eval(N, (P, Q)) :-
114:     !,
115:     eval(N, P),
116:     eval(N, Q).
117:
118: eval(N, P) :-
119:     cl(N, P, Body),
120:     eval(N, Body).
121:     %call(Z) --> Call, {translate(), cl(N, Call, B), eval(N, B)}.
122:
123:
124: %% valid(?N)
125:
126: valid(N) :-
127:     eval(N, false).
128:
129:
130: %% invalid(?N)
131:
132: invalid(N) :-
133:     \+ eval(N, false).
134:

```

```
135:
136: %% ----- Step 5 -----%%
137:
138: %% test(+N)
139:
140:
141: test(N) :-
142:     write('syllogism '), write(N), write(':'),
143:     nl,
144:     findall(Premise, p(N, Premise), PremiseList),
145:     printList(PremiseList),
146:     nl,
147:     write('    =>'),
148:     nl,
149:     c(N, Conclusion),
150:     write('    '),
151:     printListElems(Conclusion),
152:     nl,
153:     nl,
154:     write('Premises and opposite of conclusion converted to clauses:'),
155:     nl,
156:     show_clauses(N),
157:     nl,
158:     print_validity(N).
159:
160: printList([X,Y]) :-
161:     write('    '),
162:     printListElems(X),!,
163:     nl,
164:     write('    '),
165:     printListElems(Y).
166:
167: printList([X]) :-
168:     printListElems(X).
169:
170: printListElems([X|Tail]) :-
171:     write(X), write(' '),
172:     printListElems(Tail).
173:
174: printListElems([]).
175:
176:
177: print_validity(N) :-
178:     valid(N),!,
179:     write('false can be derived, syllogism '), write(N), write(' is valid. '),
180:     nl.
181:
182: print_validity(N) :-
183:     write('false cannot be derived, syllogism '), write(N), write(' is invalid. '),
184:     nl.
```

```

1:
2:
3:
4: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5: %% MAC/MCSS/MRes
6: %% 531 Prolog
7: %% Assessed Exercise 5 - Syllogisms
8: %% arguments.pl (Test syllogism arguments)
9: %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10:
11:
12: p(1,[a,robin,is,a,bird]).      %   bird(X):-robin(X)
13: p(1,[no,bird,is,a,reptile]).   %   false:-bird(X),reptile(X)
14:
15:
16:
17:
18: p(2,[a,human,is,a,mammal]).    %   mammal(X):-human(X)
19: p(2,[a,mammal,is,warm_blooded]). %   warm_blooded(X):-mammal(X)
20:
21:
22:
23:
24: p(3,[a,human,is,a,mammal]).    %   mammal(X):-human(X)
25: p(3,[some,human,is,deaf]).
26:     %   human(some(human,deaf)):-true
27:     %   deaf(some(human,deaf)):-true
28:
29:
30: p(4,[a,robin,is,a,bird]).      %   bird(X):-robin(X)
31: p(4,[no,bird,is,a,reptile]).   %   false:-bird(X),reptile(X)
32:
33:
34:
35: p(5,[every,philosopher,is,a,logician]). %   logician(X):-philosopher(X)
36: p(5,[every,philosopher,is,a,professor]). %   professor(X):-philosopher(X)
37:
38:
39: p(6,[no,vegetarian,is,a,meat_eater]).   %   false:-meat_eater(X),vegetarian(X)
40: p(6,[some,vegetarian,is,a,policeman]).
41:     %   vegetarian(some(vegetarian,policeman)):-true
42:     %   policeman(some(vegetarian,policeman)):-true
43:
44:
45: p(7,[some,student,is,rich]).
46:     %   student(some(student,rich)):- true
47:     %   rich(some(student,rich)):- true
48: p(7,[some,girl,is,not,a,student]).
49:     %   girl(some(girl,not(student))) :-true
50:     %   false:-student(some(girl,not(student)))
51:
52: p(8,[some,man,is,a,clown]).
53:     %   man(some(man,clown)):-true
54:     %   clown(some(man,clown)):-true
55: p(8,[some,man,is,bald]).
56:     %   man(some(man,bald)):-true
57:     %   bald(some(man,bald)):-true
58:
59:
60:
61: c(1,[no,robin,is,a,reptile]).
62: %   Opposite is: [some,robin,is,a,reptile]).
63: %   robin(some(robin,reptile)):-true
64: %   reptile(some(robin,reptile)):-true
65: % false can be derived, syllogism is valid
66:
67:

```

```

68: c(2,[a,human,is,warm_blooded]).
69: %   Opposite is: [some,human,is,not,warm_blooded]).
70: %   human(some(human,not(warm_blooded))):-true
71: %   false:-warm_blooded(some(human,not(warm_blooded)))
72: % false can be derived, syllogism is valid
73:
74:
75: c(3,[some,mammal,is,deaf]).
76: %   Opposite is: [no,mammal,is,deaf]).
77: %   false:-mammal(X),deaf(X)
78: % false can be derived, syllogism is valid
79:
80:
81: c(4,[no,reptile,is,a,robin]).
82: %   Opposite is: [some,reptile,is,a,robin]).
83: %   robin(some(reptile,robin)):-true
84: %   reptile(some(reptile,robin)):-true
85: % false can be derived, syllogism is valid
86:
87: c(5,[every,logician,is,a,professor]).
88: %   Opposite is [some,logician,is,not,a,professor]
89: %   logician(some(logician,not(professor))):-true
90: %   false:-professor(some(logician,not(professor)))
91: % False cannot be derived, syllogism is invalid
92:
93: c(6,[some,policeman,is,not,a,meat_eater]).
94: %   Negated conclusion is: [a,policeman,is,a,meat_eater]
95: %   meat_eater(X):-policeman(X)
96: % false can be derived, syllogism is valid
97:
98:
99: c(7,[some,girl,is,not,rich]).
100: %   Opposites: [a,girl,is,rich]
101: %   rich(X):-girl(X)
102: % false cannot be derived, syllogism is invalid
103:
104: c(8,[some,clown,is,bald]).
105: %   Opposite is [no,clown,is,bald]
106: %   false:-clown(X),bald(X)
107: % false cannot be derived, syllogism is invalid

```

Prolog Syllogisms/Full Tests

TestLog.txt: 1/2

Noah-Vincenz Noeh - nn4718: a5

```

1: % compiling /root/labcat/labcat/engines/lib/prolog/automarker.pl...
2: % loading /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/timeout.po...
3: % module timeout imported into user
4: % loading /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/types.po...
5: % module types imported into timeout
6: % loaded /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/types.po in module types, 0 msec 4112 bytes
7: % loading foreign resource /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/x86_64-linux-glibc2.17/timeout.so in module timeout
8: % loaded /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/timeout.po in module timeout, 0 msec 51056 bytes
9: % loading /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/file_systems.po...
10: % module file_systems imported into user
11: % module types imported into file_systems
12: % loading /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/system.po...
13: % module system imported into file_systems
14: % module types imported into system
15: % loaded /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/system.po in module system, 10 msec 4720 bytes
16: % loading /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/lists.po...
17: % module lists imported into file_systems
18: % module types imported into lists
19: % loaded /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/lists.po in module lists, 0 msec 107408 bytes
20: % loaded /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/file_systems.po in module file_systems, 10 msec 165488 bytes
21: % compiled /root/labcat/labcat/engines/lib/prolog/automarker.pl in module user, 23 0 msec 1220880 bytes
22: SICStus 4.3.5 (x86_64-linux-glibc2.17): Tue Dec 6 10:41:06 PST 2016
23: Licensed to SP4.3doc.ic.ac.uk
24: % compiling /tmp/d20181122-38-97zyad/src/solution.pl...
25: % compiled /tmp/d20181122-38-97zyad/src/solution.pl in module autom, 10 msec 30480 bytes
26: yes
27: % compiling /tmp/d20181122-38-97zyad/src/utilities.pl...
28: % compiled /tmp/d20181122-38-97zyad/src/utilities.pl in module autom, 0 msec 2512 bytes
29: yes
30: % compiling /tmp/d20181122-38-97zyad/src/private_test_data.pl...
31: % compiled /tmp/d20181122-38-97zyad/src/private_test_data.pl in module autom, 10 msec 3024 bytes
32: yes
33: % compiling /tmp/d20181122-38-97zyad/src/arguments.pl...
34: % compiled /tmp/d20181122-38-97zyad/src/arguments.pl in module autom, 0 msec 7856 bytes
35: yes
36: % compiling /tmp/d20181122-38-97zyad/src/arguments.pl...
37: % compiled /tmp/d20181122-38-97zyad/src/arguments.pl in module submission, 0 msec 7440 bytes
38: * /tmp/d20181122-38-97zyad/src/arguments.pl is previously loaded into module autom
39: * now it is reloaded into module submission
40: yes
41: % compiling /tmp/d20181122-38-97zyad/src/utilities.pl...
42: % compiled /tmp/d20181122-38-97zyad/src/utilities.pl in module submission, 0 msec 3488 bytes
43: * /tmp/d20181122-38-97zyad/src/utilities.pl is previously loaded into module autom
44: * now it is reloaded into module submission
45: yes
46: % compiling /tmp/d20181122-38-97zyad/src/syllogisms.pl...
47: % including /tmp/d20181122-38-97zyad/src/arguments.pl...
48: The procedure p/2 is being redefined.
49: Old file: /tmp/d20181122-38-97zyad/src/arguments.pl
50: New file: /tmp/d20181122-38-97zyad/src/syllogisms.pl
51: Do you really want to redefine it? (y, n, p, s, a, b, or ?)
52: y redefine this procedure
53: n don't redefine this procedure

```

```

54: p redefine this procedure and don't ask again
55: s don't redefine this procedure and don't ask again
56: a abort
57: b break
58: ? print this information
59: Do you really want to redefine it? (y, n, p, s, a, b, or ?)
60: The procedure c/2 is being redefined.
61: Old file: /tmp/d20181122-38-97zyad/src/arguments.pl
62: New file: /tmp/d20181122-38-97zyad/src/syllogisms.pl
63: Do you really want to redefine it? (y, n, p, s, a, b, or ?)
64: y redefine this procedure
65: n don't redefine this procedure
66: p redefine this procedure and don't ask again
67: s don't redefine this procedure and don't ask again
68: a abort
69: b break
70: ? print this information
71: Do you really want to redefine it? (y, n, p, s, a, b, or ?)
72: y redefine this procedure
73: n don't redefine this procedure
74: p redefine this procedure and don't ask again
75: s don't redefine this procedure and don't ask again
76: a abort
77: b break
78: ? print this information
79: Do you really want to redefine it? (y, n, p, s, a, b, or ?)
80: y redefine this procedure
81: n don't redefine this procedure
82: p redefine this procedure and don't ask again
83: s don't redefine this procedure and don't ask again
84: a abort
85: b break
86: ? print this information
87: Do you really want to redefine it? (y, n, p, s, a, b, or ?)
88: % included /tmp/d20181122-38-97zyad/src/arguments.pl in module submission, 0 msec 33472 bytes
89: % including /tmp/d20181122-38-97zyad/src/utilities.pl...
90: The procedure forall/2 is being redefined.
91: Old file: /tmp/d20181122-38-97zyad/src/utilities.pl
92: New file: /tmp/d20181122-38-97zyad/src/syllogisms.pl
93: Do you really want to redefine it? (y, n, p, s, a, b, or ?)
94: y redefine this procedure
95: n don't redefine this procedure
96: p redefine this procedure and don't ask again
97: s don't redefine this procedure and don't ask again
98: a abort
99: b break
100: ? print this information
101: Do you really want to redefine it? (y, n, p, s, a, b, or ?)
102: y redefine this procedure
103: n don't redefine this procedure
104: p redefine this procedure and don't ask again
105: s don't redefine this procedure and don't ask again
106: a abort
107: b break
108: ? print this information
109: Do you really want to redefine it? (y, n, p, s, a, b, or ?)
110: y redefine this procedure
111: n don't redefine this procedure
112: p redefine this procedure and don't ask again
113: s don't redefine this procedure and don't ask again
114: a abort
115: b break
116: ? print this information
117: Do you really want to redefine it? (y, n, p, s, a, b, or ?)
118: y redefine this procedure
119: n don't redefine this procedure

```

```
120:      p      redefine this procedure and don't ask again
121:      s      don't redefine this procedure and don't ask again
122:      a      abort
123:      b      break
124:      ?      print this information
125: Do you really want to redefine it? (y, n, p, s, a, b, or ?)
126:      y      redefine this procedure
127:      n      don't redefine this procedure
128:      p      redefine this procedure and don't ask again
129:      s      don't redefine this procedure and don't ask again
130:      a      abort
131:      b      break
132:      ?      print this information
133: Do you really want to redefine it? (y, n, p, s, a, b, or ?)
134: % included /tmp/d20181122-38-97zyad/src/utilities.pl in module submission, 0 msec
1472 bytes
135: % compiled /tmp/d20181122-38-97zyad/src/syllogisms.pl in module submission, 20 mse
c 55760 bytes
136: yes
137: yes
138: yes
139: yes
140: yes
141: yes
142: yes
143: yes
144: yes
145: yes
146: yes
147: yes
148: yes
149: yes
150: yes
151: yes
152: yes
153: yes
154: yes
```

```

1: ----- Test 1 :: Opposite -----
2:
3: | ? find Opp: opposite([a,cat,is,a,mammal],Opp).
4:   Opp = [some,cat,is,not,a,mammal] ;           %% correct
5:
6: ----- Test 2 :: Opposite -----
7:
8: | ? find Opp: opposite([every,cat,is,a,mammal],Opp).
9:   Opp = [some,cat,is,not,a,mammal] ;           %% correct
10:
11: ----- Test 3 :: Opposite -----
12:
13: | ? find Opp: opposite([some,cat,is,a,mammal],Opp).
14:   Opp = [no,cat,is,a,mammal] ;                 %% correct
15:
16: ----- Test 4 :: Opposite -----
17:
18: | ? find Opp: opposite([some,cat,is,not,a,mammal],Opp).
19:   Opp = [a,cat,is,a,mammal] ;                   %% correct
20:
21: ----- Test 5 :: Opposite -----
22:
23: | ? find Opp: opposite([a,banker,is,greedy],Opp).
24:   Opp = [some,banker,is,not,greedy] ;           %% correct
25:
26: ----- Test 6 :: Opposite -----
27:
28: | ? find Opp: opposite([every,banker,is,greedy],Opp).
29:   Opp = [some,banker,is,not,greedy] ;           %% correct
30:
31: ----- Test 7 :: Opposite -----
32:
33: | ? find Opp: opposite([no,banker,is,greedy],Opp).
34:   Opp = [some,banker,is,greedy] ;               %% correct
35:
36: ----- Test 8 :: Opposite -----
37:
38: | ? find Opp: opposite([some,banker,is,not,greedy],Opp).
39:   Opp = [a,banker,is,greedy] ;                   %% correct
40:
41: =====
42: Step 2 (syllogism/1)
43:
44: ----- Test 9 :: Parsing Syllogisms -----
45:
46: | ? find Clauses: syllogism(Clauses,[a,cat,is,a,mammal],[]).
47:   Clauses = [(mammal(_3173):-cat(_3173))] ;
48: No more solutions      (All correct)
49: No missing solutions
50:
51: ----- Test 10 :: Parsing Syllogisms -----
52:
53: | ? find Clauses: syllogism(Clauses,[every,cat,is,a,mammal],[]).
54:   Clauses = [(mammal(_3101):-cat(_3101))] ;
55: No more solutions      (All correct)
56: No missing solutions
57:
58: ----- Test 11 :: Parsing Syllogisms -----
59:
60: | ? find Clauses: syllogism(Clauses,[some,cat,is,a,mammal],[]).
61:   Clauses = [(cat(some(cat,mammal)):-true),(mammal(some(cat,mammal)):-true)] ;
62: No more solutions      (All correct)
63: No missing solutions
64:
65: ----- Test 12 :: Parsing Syllogisms -----
66:
67: | ? find Clauses: syllogism(Clauses,[some,cat,is,not,a,mammal],[]).

```

```

68:   Clauses = [(cat(some(cat,not(mammal))):-true),(false:-mammal(some(cat,not(mammal))))] ;
69: No more solutions      (All correct)
70: No missing solutions
71:
72: ----- Test 13 :: Parsing Syllogisms -----
73:
74: | ? find Clauses: syllogism(Clauses,[no,banker,is,greedy],[]).
75:   Clauses = [(false:-banker(_3097),greedy(_3097))] ;
76: No more solutions      (All correct)
77: No missing solutions
78:
79: =====
80: Step 2 (syllogism/1)
81:
82: TESTS PASSED: 13 / 13 ; MARKS: 38 / 38
83: =====
84:
85: =====
86: Step 3 (translate/1, cl/3)
87:
88:
89: About to call translate(1) ... [[a,robin,is,a,bird],[no,bird,is,a,reptile]]
90: [(false:-bird(_4227),reptile(_4227)),(bird(_4109):-robin(_4109)),(robin(some(robin,reptile)):-true),(reptile(some(robin,reptile)):-true)]done
91: About to call translate(2) ... [[a,human,is,a,mammal],[a,mammal,is,warm_blooded]]
92: [(warm_blooded(_4121):-mammal(_4121)),(mammal(_4003):-human(_4003)),(human(some(human,not(warm_blooded))):-true),(false:-warm_blooded(some(human,not(warm_blooded))))]done
93: About to call translate(3) ... [[a,human,is,a,mammal],[some,human,is,deaf]]
94: [(human(some(human,deaf)):-true),(deaf(some(human,deaf)):-true),(mammal(_3945):-human(_3945)),(false:-mammal(_3821),deaf(_3821))]done
95: About to call translate(4) ... [[a,robin,is,a,bird],[no,bird,is,a,reptile]]
96: [(false:-bird(_4109),reptile(_4109)),(bird(_3991):-robin(_3991)),(reptile(some(robin,robin)):-true),(robin(some(reptile,robin)):-true)]done
97: About to call translate(5) ... [[every,philosopher,is,a,logician],[every,philosopher,is,a,professor]]
98: [(professor(_4143):-philosopher(_4143)),(logician(_4025):-philosopher(_4025)),(logician(some(logician,not(professor))):-true),(false:-professor(some(logician,not(professor))))]done
99: About to call translate(6) ... [[no,vegetarian,is,a,meat_eater],[some,vegetarian,is,a,policeman]]
100: [(vegetarian(some(vegetarian,policeman)):-true),(policeman(some(vegetarian,policeman)):-true),(false:-vegetarian(_3985),meat_eater(_3985)),(meat_eater(_3867):-policeman(_3867))]done
101: About to call translate(7) ... [[some,student,is,rich],[some,girl,is,not,a,student]]
102: [(girl(some(girl,not(student))):-true),(false:-student(some(girl,not(student))))),(student(some(student,rich)):-true),(rich(some(student,rich)):-true),(rich(_3851):-girl(_3851))]done
103: About to call translate(8) ... [[some,man,is,a,clown],[some,man,is,bald]]
104: [(man(some(man,bald)):-true),(bald(some(man,bald)):-true),(man(some(man,clown)):-true),(clown(some(man,clown)):-true),(false:-clown(_3821),bald(_3821))]done
105:
106: ----- Test 1 :: Asserting Translations -----
107:
108: | ? find Cl: cl(1,_1185,_1225),Cl=(_1185:-_1225).
109:   Cl = false:-bird(_3611),reptile(_3611) ;
110:   Cl = bird(_3605):-robin(_3605) ;
111:   Cl = robin(some(robin,reptile)):-true ;
112:   Cl = reptile(some(robin,reptile)):-true ;
113: No more solutions      (All correct)
114: No missing solutions
115:
116: ----- Test 2 :: Asserting Translations -----
117:
118: | ? find Cl: cl(2,_1185,_1225),Cl=(_1185:-_1225).
119:   Cl = warm_blooded(_3643):-mammal(_3643) ;

```



```

120: C1 = mammal(_3643):-human(_3643) ;
121: C1 = human(some(human,not(warm_blooded))):-true ;
122: C1 = false:-warm_blooded(some(human,not(warm_blooded))) ;
123: No more solutions      (All correct)
124: No missing solutions
125:
126: ----- Test 3 :: Asserting Translations -----
127:
128: | ? find C1: cl(3,_1185,_1225),C1=(_1185:-_1225).
129: C1 = human(some(human,deaf)):-true ;
130: C1 = deaf(some(human,deaf)):-true ;
131: C1 = mammal(_3643):-human(_3643) ;
132: C1 = false:-mammal(_3649),deaf(_3649) ;
133: No more solutions      (All correct)
134: No missing solutions
135:
136: ----- Test 4 :: Asserting Translations -----
137:
138: | ? find C1: cl(4,_1185,_1225),C1=(_1185:-_1225).
139: C1 = false:-bird(_3653),reptile(_3653) ;
140: C1 = bird(_3647):-robin(_3647) ;
141: C1 = reptile(some(reptile,robin)):-true ;
142: C1 = robin(some(reptile,robin)):-true ;
143: No more solutions      (All correct)
144: No missing solutions
145:
146: ----- Test 5 :: Asserting Translations -----
147:
148: | ? find C1: cl(5,_1185,_1225),C1=(_1185:-_1225).
149: C1 = professor(_3647):-philosopher(_3647) ;
150: C1 = logician(_3647):-philosopher(_3647) ;
151: C1 = logician(some(logician,not(professor))):-true ;
152: C1 = false:-professor(some(logician,not(professor))) ;
153: No more solutions      (All correct)
154: No missing solutions
155:
156: ----- Test 6 :: Asserting Translations -----
157:
158: | ? find C1: cl(6,_1185,_1225),C1=(_1185:-_1225).
159: C1 = vegetarian(some(vegetarian,policeman)):-true ;
160: C1 = policeman(some(vegetarian,policeman)):-true ;
161: C1 = false:-vegetarian(_3657),meat_eater(_3657) ;
162: C1 = meat_eater(_3651):-policeman(_3651) ;
163: No more solutions      (All correct)
164: No missing solutions
165:
166: ----- Test 7 :: Asserting Translations -----
167:
168: | ? find C1: cl(7,_1185,_1225),C1=(_1185:-_1225).
169: C1 = girl(some(girl,not(student))):-true ;
170: C1 = false:-student(some(girl,not(student))) ;
171: C1 = student(some(student,rich)):-true ;
172: C1 = rich(some(student,rich)):-true ;
173: C1 = rich(_3647):-girl(_3647) ;
174: No more solutions      (All correct)
175: No missing solutions
176:
177: ----- Test 8 :: Asserting Translations -----
178:
179: | ? find C1: cl(8,_1185,_1225),C1=(_1185:-_1225).
180: C1 = man(some(man,bald)):-true ;
181: C1 = bald(some(man,bald)):-true ;
182: C1 = man(some(man,clown)):-true ;
183: C1 = clown(some(man,clown)):-true ;
184: C1 = false:-clown(_3649),bald(_3649) ;
185: No more solutions      (All correct)
186: No missing solutions

```

```

187:
188:
189: =====
190: Step 3 (translate/1, cl/3)
191: TESTS PASSED: 8 / 8 ; MARKS: 12 / 12
192: =====
193:
194: =====
195: Step 4 (eval/2, valid/1, invalid/1)
196:
197: ----- Test 1 :: Evaluate An Argument -----
198:
199: | ? eval(1,false).
200: yes      %% correct
201:
202: ----- Test 2 :: Evaluate An Argument -----
203:
204: | ? eval(2,false).
205: yes      %% correct
206:
207: ----- Test 3 :: Evaluate An Argument -----
208:
209: | ? eval(3,false).
210: yes      %% correct
211:
212: ----- Test 4 :: Evaluate An Argument -----
213:
214: | ? eval(4,false).
215: yes      %% correct
216:
217: ----- Test 5 :: Evaluate An Argument -----
218:
219: | ? eval(5,false).
220: no       %% correct
221:
222: ----- Test 6 :: Evaluate An Argument -----
223:
224: | ? eval(6,false).
225: yes      %% correct
226:
227: ----- Test 7 :: Evaluate An Argument -----
228:
229: | ? eval(7,false).
230: no       %% correct
231:
232: ----- Test 8 :: Evaluate An Argument -----
233:
234: | ? eval(8,false).
235: no       %% correct
236:
237: ----- Test 9 :: Find 5 Valid -----
238:
239: | ? find N: valid(N).
240: N = 1 ;
241: N = 2 ;
242: N = 3 ;
243: N = 4 ;
244: N = 6 ;
245: No more solutions      (All correct)
246: No missing solutions
247:
248: ----- Test 10 :: Valid? Not an Argument -----
249:
250: | ? valid(p0).
251: no       %% correct
252:
253: ----- Test 11 :: Find 3 Invalid -----

```

```

254:
255: | ? find N: invalid(N).
256: No solution          %% WRONG
257:
258: Model answer produces the following solutions:
259:   N = 5 ;
260:   N = 7 ;
261:   N = 8 ;
262: Model solutions end
263:
264: ----- Test 12 :: Invalid? Not an Argument -----
265:
266: | ? invalid(p0).
267: yes          %% WRONG
268:
269:
270: =====
271: Step 4 (eval/2, valid/1, invalid/1)
272: TESTS PASSED: 10 / 12 ; MARKS: 13 / 17
273: =====
274:
275: =====
276: Step 5 (test/1)
277:
278: ----- Test 1 :: Evaluate and Output Argument -----
279:
280: syllogism 1:
281:   a robin is a bird
282:   no bird is a reptile
283:   =>
284:   no robin is a reptile
285:
286: Premises and opposite of conclusion converted to clauses:
287:   false:-bird(_3053),reptile(_3053).
288:   bird(_3047):-robin(_3047).
289:   robin(some(robin,reptile)):-true.
290:   reptile(some(robin,reptile)):-true.
291:
292: false can be derived, syllogism 1 is valid.
293:
294:
295: ----- Test 2 :: Evaluate and Output Argument -----
296:
297: syllogism 2:
298:   a human is a mammal
299:   a mammal is warm_blooded
300:   =>
301:   a human is warm_blooded
302:
303: Premises and opposite of conclusion converted to clauses:
304:   warm_blooded(_2983):-mammal(_2983).
305:   mammal(_2983):-human(_2983).
306:   human(some(human,not(warm_blooded))):-true.
307:   false:-warm_blooded(some(human,not(warm_blooded))).
308:
309: false can be derived, syllogism 2 is valid.
310:
311:
312: ----- Test 3 :: Evaluate and Output Argument -----
313:
314: syllogism 3:
315:   a human is a mammal
316:   some human is deaf
317:   =>
318:   some mammal is deaf
319:
320: Premises and opposite of conclusion converted to clauses:

```

```

321:   human(some(human,deaf)):-true.
322:   deaf(some(human,deaf)):-true.
323:   mammal(_2983):-human(_2983).
324:   false:-mammal(_2989),deaf(_2989).
325:
326: false can be derived, syllogism 3 is valid.
327:
328:
329: ----- Test 4 :: Evaluate and Output Argument -----
330:
331: syllogism 4:
332:   a robin is a bird
333:   no bird is a reptile
334:   =>
335:   no reptile is a robin
336:
337: Premises and opposite of conclusion converted to clauses:
338:   false:-bird(_3017),reptile(_3017).
339:   bird(_3011):-robin(_3011).
340:   reptile(some(reptile,robin)):-true.
341:   robin(some(reptile,robin)):-true.
342:
343: false can be derived, syllogism 4 is valid.
344:
345:
346: ----- Test 5 :: Evaluate and Output Argument -----
347:
348: syllogism 5:
349:   every philosopher is a logician
350:   every philosopher is a professor
351:   =>
352:   every logician is a professor
353:
354: Premises and opposite of conclusion converted to clauses:
355:   professor(_3011):-philosopher(_3011).
356:   logician(_3011):-philosopher(_3011).
357:   logician(some(logician,not(professor))):-true.
358:   false:-professor(some(logician,not(professor))).
359:
360: false cannot be derived, syllogism 5 is invalid.
361:
362:
363: ----- Test 6 :: Evaluate and Output Argument -----
364:
365: syllogism 6:
366:   no vegetarian is a meat_eater
367:   some vegetarian is a policeman
368:   =>
369:   some policeman is not a meat_eater
370:
371: Premises and opposite of conclusion converted to clauses:
372:   vegetarian(some(vegetarian,policeman)):-true.
373:   policeman(some(vegetarian,policeman)):-true.
374:   false:-vegetarian(_3033),meat_eater(_3033).
375:   meat_eater(_3027):-policeman(_3027).
376:
377: false can be derived, syllogism 6 is valid.
378:
379:
380: ----- Test 7 :: Evaluate and Output Argument -----
381:
382: syllogism 7:
383:   some student is rich
384:   some girl is not a student
385:   =>
386:   some girl is not rich
387:

```

```
388: Premises and opposite of conclusion converted to clauses:
389:   girl(some(girl,not(student))):-true.
390:   false:-student(some(girl,not(student))).
391:   student(some(student,rich)):-true.
392:   rich(some(student,rich)):-true.
393:   rich(_3011):-girl(_3011).
394:
395: false cannot be derived, syllogism 7 is invalid.
396:
397:
398: ----- Test 8 :: Evaluate and Output Argument -----
399:
400: syllogism 8:
401:   some man is a clown
402:   some man is bald
403:   =>
404:   some clown is bald
405:
406: Premises and opposite of conclusion converted to clauses:
407:   man(some(man,bald)):-true.
408:   bald(some(man,bald)):-true.
409:   man(some(man,clown)):-true.
410:   clown(some(man,clown)):-true.
411:   false:-clown(_2989),bald(_2989).
412:
413: false cannot be derived, syllogism 8 is invalid.
414:
415:
416:
417: =====
418: Step 5 (test/1)
419: TESTS PASSED:  8 / 8 ; MARKS:  8 / 8
420: =====
421:
422:
423: ===== SUMMARY (???) =====
424:
425: TESTS PASSED:  39 / 41 ; MARKS:  71 / 75
426:
427: =====
```